

## **КОМПЬЮТЕРНОЕ МОДЕЛИРОВАНИЕ ИДЕНТИФИКАЦИИ ЛИЧНОСТИ ПО РАДУЖНОЙ ОБОЛОЧКЕ ГЛАЗА НА ОСНОВЕ OPENCV**

**Т.Ю. Назарова**

студент, e-mail: tanyanazarova-94@yandex.ru

**Д.Н. Лавров**

к.т.н., доцент, e-mail: lavrov@omsu.ru

Омский государственный университет им. Ф.М. Достоевского, Омск, Россия

**Аннотация.** В статье представлены результаты конструирования алгоритма распознавания на основе использования библиотеки OpenCV. Представлены результаты первичных экспериментов по распознаванию на небольшой базе изображений.

**Ключевые слова:** радужная оболочка глаза, идентификация, распознавание.

### **Введение**

Целью нашей работы является построение системы компьютерного моделирования распознавания личности по радужной оболочке глаза. Работа является продолжением работ [1, 2]

На данный момент существуют различные методы распознавания образов. Все они разделяются на несколько этапов (рис. 1):

1. Получение биометрических данных.
2. Предварительная обработка.
3. Извлечение признаков из биометрических данных.
4. Классификация биометрических образцов (идентификация) или сравнение с образцом (верификация).

Перечислим основные приёмы, используемые на каждом этапе. Рассмотрение этапа получения изображений выходит за рамки данной статьи.

### **1. Предварительная обработка изображения глаза**

Этапы предварительной обработки изображения для поиска границ радужной оболочки глаза выглядят следующим образом: фильтрация изображения, улучшение контрастности, бинаризация изображения и выделение границ.

**Методы препроцессинга.** Фильтрация используется для уменьшения шума на изображении. Рассмотрим несколько наиболее часто используемых приёмов.

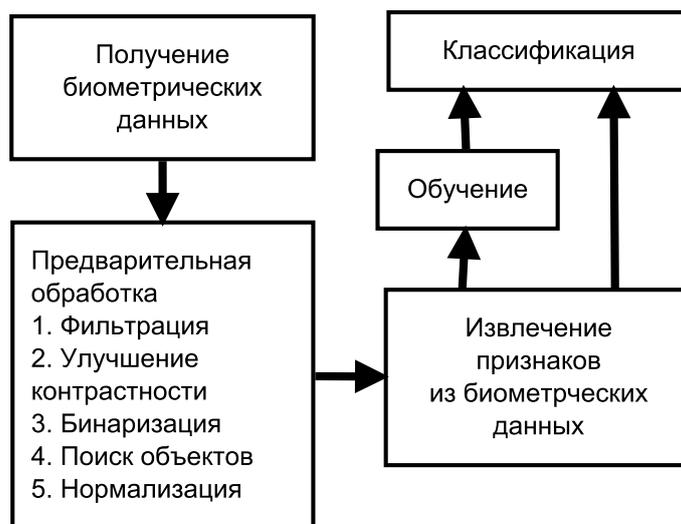


Рис. 1. Архитектура системы распознавания человека по радужной оболочке глаза

*Размытие Гаусса* — это фильтр, построенный на использовании нормального распределения для построения ядра свёртки  $G(x, y) = \frac{1}{\sqrt{2\pi\sigma^2}} e^{-\frac{x^2+y^2}{2\sigma^2}}$ . При-

мер ядра свёртки  $3 \times 3$  для фильтра:  $K(x, y) = \frac{1}{16} \begin{bmatrix} 1 & 2 & 1 \\ 2 & 4 & 2 \\ 1 & 2 & 1 \end{bmatrix}$ ,  $x = -1, 0, 1$ ;  
 $y = -1, 0, 1$ .

Вычисление интенсивности пикселя производится по формуле вычисления свёртки  $I(x_i, y_i) = \sum_{x=-\lfloor n/2 \rfloor}^{\lfloor n/2 \rfloor} \sum_{y=-\lfloor n/2 \rfloor}^{\lfloor n/2 \rfloor} K(x, y) \cdot I(x_i + x, y_i + y)$ .

*Медианный фильтр.* Медиана рассчитывается путём сортировки всех значений пикселей окрестности рассматриваемого пикселя в порядке возрастания и выбора интенсивности в середине отсортированного списка. Затем интенсивность рассматриваемого пикселя заменяется на медианное значение соседей.

Для повышения контраста используется алгоритм *контрастного ограниченного адаптивного выравнивания гистограммы* (CLAHE) [7].

Последовательность действий этого алгоритма следующая: 1) вычисляется гистограмма  $H_{(x_i, y_i), k}(l)$  окрестности пикселя с координатами  $(x_i, y_i)$ , здесь  $l$  — интенсивность, а  $k \times k$  — размер рассматриваемой окрестности; 2) гистограмма обрезается по заданному порогу; 3) срезанные значения яркости равномерно перераспределяются (это можно сделать итеративно, пока не останется ни одного пикселя с интенсивностью выше ранее заданного порога); 4) по пересчитанной гистограмме  $\hat{H}_{(x_i, y_i), k}(l)$  вычисляется кумулятивная функция распределения:  $CDF_{(x_i, y_i), k}(l) = \sum_{i=0}^l \hat{H}_{(x_i, y_i), k}(i)$ ; 5) затем вычисляется интенсивность пикселя  $I'(x_i, y_i) = 255 \cdot CDF_{(x_i, y_i), k}(I(x_i, y_i))$  [6].

*Бинаризация* — алгоритм преобразования полноцветного изображения в монохромное. Бинаризация бывает двух видов — пороговая и адаптивная. Пороговая бинаризация — метод, при котором выбирается порог бинаризации, и, если выбранный пиксель выше этого порога, то он становится белым, ниже — чёрным. Адаптивные методы работают с участками изображения и используются при неоднородном освещении объектов [3].

*Преобразование Хафа* предназначено для поиска объектов (в данном случае окружностей), принадлежащих определённому классу фигур с использованием процедуры голосования. Процедура голосования применяется к пространству параметров, из которого и получаются объекты определённого класса фигур по локальному максимуму в аккумуляторном пространстве, которое строится при вычислении преобразования Хафа. Преобразование Хафа основывается на представлении искомого объекта в виде параметрического уравнения:  $(x - a)^2 + (y - b)^2 = r^2$ . Параметры этого уравнения представляют фазовое пространство. Перебираются все точки границ, и делается предположение, что точка принадлежит линии искомого объекта. Таким образом, для каждой точки изображения рассчитывается нужное уравнение и получаются необходимые параметры, которые сохраняются в пространстве Хафа. Финальным шагом является обход пространства Хафа и выбор максимальных значений, за которые «проголосовало» больше всего пикселей картинки, что и даёт нам параметры для уравнений искомого объекта [4].

*Нормализация* изображения нужна для того, чтобы каждое изображение имело единый размер. В противном случае возникают трудности при сравнении радужных оболочек глаза.

Даугман предлагает перевести изображение из декартовой системы координат  $(x, y)$  в *полярную систему координат*  $(\rho, \varphi)$ . Эта форма представления радужной оболочки (*развёртка*) имеет ряд преимуществ по сравнению с кольцом: поворот изображения заменяется циклическим сдвигом.

Для выделения существенных признаков на радужной оболочке используется *сингулярное разложение*. Представим, что развёртка является матрицей. Для любой вещественной  $(m \times m)$  матрицы  $A$  существуют две вещественные ортогональные матрицы  $U$  и  $V$  такие, что  $\Sigma = U^T A V$ , где диагональная матрица  $\Sigma$ . Матрицы  $U$  и  $V$  выбираются так, чтобы диагональные элементы матрицы  $\Sigma$  имели вид:  $\sigma_1 \geq \sigma_2 \geq \dots \geq \sigma_r$ , где  $r$  — ранг матрицы  $A$ . Следующим шагом необходимо удалить малозначимые (определяется по порогу) диагональные элементы матрицы  $\Sigma$  и обратным преобразованием восстановить исходное изображение, в котором останутся только главные компоненты.

Независимо от нашей разработки аналогичные подходы к предобработке были использованы в докторской диссертации Матвеева И.А. «Методы и алгоритмы автоматической обработки изображений радужной оболочки глаза». В работе исследованы характеристики изображений глаз человека, рассмотрены отдельные методы для выделения границ радужной оболочки, на основании полученных характеристик методы и алгоритмы были сведены в единую систему и протестированы [5].

## 2. Методы идентификации биометрических образцов

Классификация биометрических образцов происходит по следующему алгоритму:

1. Выделение признаков у биометрического образца.
2. Сравнение полученного вектора признаков с векторами, которые находятся в базе.
3. Подтверждение гипотезы о совпадении текущего биометрического образца с одним из образцов в базе.

Рассмотрим несколько алгоритмов, идентифицирующих биометрические образцы. Метод опорных векторов направлен на перевод исходного вектора в пространство более высокой размерности и поиск гиперплоскости, которая разделит два класса в этом пространстве. Размерность гиперплоскости на единицу меньше, чем размерность исходного пространства.

*Метод опорных векторов (SVM)* направлен на перевод исходного вектора в пространство более высокой размерности и поиск гиперплоскости, которая разделит два класса в этом пространстве. Размерность гиперплоскости на единицу меньше, чем размерность исходного пространства.

Рассмотрим бинарную задачу классификации. Пусть имеется обучающая выборка:  $(x_1, y_1), \dots, (x_m, y_m)$ ,  $x_i \in R^n$ ,  $y_i \in \{-1, 1\}$ . SVM строит функцию классификации  $F(x) = \text{sign}(\langle w, x \rangle + b)$ , где  $\langle w, x \rangle$  — скалярное произведение,  $w$  — вектор нормали к оптимальной гиперплоскости,  $b$  — дополнительный параметр сдвига. Если  $F(x) = 1$ , то объект попадает в один класс, а если  $F(x) = -1$ , то объект попадает в другой класс. Необходимо выбрать такие параметры  $w$  и  $b$ , которые расположат гиперплоскость на максимальное расстояние от двух классов. Предположим, что расстояние от каждого класса должно быть равным. Задача вычисления максимума эквивалентна задаче нахождения минимума, которую можно описать в виде следующих формул:

$$\begin{cases} \arg \min_{w, b} \|w\|^2 \\ y_i(\langle w, x \rangle + b) \geq 1, \quad i = 1 \dots m. \end{cases}$$

На практике довольно редко можно разделить два класса гиперплоскостью, поэтому важен случай линейной неразделимости.

В этом случае обучающую выборку переводят в пространство более высокой размерности с помощью следующего отображения:  $\varphi : R^n \rightarrow x$ . При этом отображение выбирается таким образом, чтобы разделить классы, выборка становится линейно разделима. Каждое скалярное произведение в приведённых выше формулах заменяется нелинейной функцией ядра. Ядром классификатора называется функция  $k(x, x') = \langle \varphi(x), \varphi(x') \rangle$ . Ядром может служить любая положительная симметричная функция от двух переменных. Она должна быть положительной, чтобы была возможность посчитать минимум  $\|w\|^2$  [6].

Следующий метод классификации — это *нейронные сети*. Они представляют собой структуру из нескольких слоев, содержащих вычислительные элементы (нейроны) и связи между ними. Искусственный нейрон работает по следующему принципу:

1. На каждый элемент входного слоя попадает входной сигнал  $x_i$ , который умножается на определённый вес  $w_i$ , соответствующий этому сигналу.
2. Все полученные результаты суммируются. Эта взвешенная сумма представляет степень возбуждения нейрона.
3. Взвешенная сумма передаётся в функцию активации  $\varphi$ , результат которой является выходным сигналом нейрона. Функция  $\varphi$  бывает разных типов:
  - пороговая функция, передающая на выход единичный сигнал;
  - сигмоидальная функция, которая преобразует входящие сигналы в значение в диапазоне  $\{0; 1\}$ ;
  - гиперболический тангенс, который позволяет получить значения разных знаков.

Для задач кластеризации и распознавания используются сети прямого пространства, в которых сигнал распространяется строго от входного слоя к выходному, сигнал в обратном направлении не распространяется.

Для распознавания человека по радужной оболочке глаза необходимо применять обучение нейронной сети на основе обучающей выборки, которая содержит в себе входные значения и правильные выходные значения. Для проверки правильности обучения также имеется тестовая выборка, по которой производится оценка качества сети.

*Метод Байеса* основан на предположении, что расположение интенсивностей на радужной оболочке глаза подчиняется нормальному распределению. Интенсивности  $I(x)$  одного образца представляют собой вектор.

Выбирается обучающая база, состоящая из  $N$ -векторов. Для этих векторов рассчитывается их среднее значение по формуле:

$$\bar{x} = \frac{1}{N} \sum_{i=1}^N x_i.$$

На следующем шаге необходимо построить матрицу корреляции:

$$\hat{R} = \frac{1}{N-1} \sum_{i=1}^N (x_i - \bar{x}_i)(x_i - \bar{x}_i)^T.$$

Плотность нормального распределения в общем виде выглядит таким образом:

$$f(x) = \frac{1}{(2\pi)^n \det(R)} e^{-\frac{1}{2}(x_i - \bar{x}_i)^T R^{-1} (x_i - \bar{x}_i)}.$$

Эллипс, на котором показатель степени в последней формуле, а значит и плотность распределения, имеют постоянное значение, называется эллипсом равной плотности, ограниченная им область — эллипсом рассеивания,  $\bar{x}_i$  — центр рассеивания. После расчёта среднего значения и матрицы корреляции выбирается некоторый порог для определения, попадает ли наш исходный вектор  $x_i$  внутрь эллипса рассеивания. Значение  $P$  выбирается исходя из правила трёх сигм — практически все значения нормально распределённой случайной величины лежат в интервале  $(\bar{x} - \delta; \bar{x} + \delta)$ . Более строго — приблизительно с

вероятностью 0,9973 значение нормально распределённой случайной величины лежит в указанном интервале.

Метод, основанный на расстоянии Хэмминга, предполагает, что все биологические образцы можно представить в виде битовой последовательности. Для этого необходимо провести бинаризацию всего экземпляра при определённом пороге бинаризации (например, равном 127).

*Расстояние Хэмминга* — это число бит, в которых два битовых кода различны:

$$d = W(Z) = W(X \text{ xor } Y),$$

где  $X$  и  $Y$  — битовые коды. Если рассматривать два битовых кода, полученных с одной и той же радужной оболочки, то расстояние Хэмминга будет близко к 0.

В случае метода, основанного на использовании скалярного произведения, вектор представляет собой пиксели изображения в оттенках серого. Производится вычисление косинусов углов между входным вектором и векторами, находящимися в базе, по следующей формуле:

$$\cos(\widehat{a, b}) = \frac{\langle a, b \rangle}{|a| \cdot |b|}.$$

Если полученное значение равно 1, то векторы  $a$  и  $b$  находятся в линейной зависимости, следовательно, принадлежат одному объекту. Результатом алгоритма является вектор, косинус которого с исходным вектором ближе всего к 1.

Метод, основанный на коэффициенте *корреляции Спирмена*, является ранговым, то есть для оценки силы связи используются не численные значения, а соответствующие им ранги. Коэффициент инвариантен по отношению к любому монотонному преобразованию шкалы измерения.

Пусть заданы две выборки  $X = (x_1, \dots, x_n)$ ,  $Y = (y_1, \dots, y_n)$ . Коэффициент *корреляции Спирмена* вычисляется по формуле:

$$\rho = 1 - \frac{6}{n(n+1)} \sum_{i=1}^n (R_i - S_i)^2,$$

где  $R_i$  — ранг наблюдения  $x_i$  в ряду  $X$ ,  $S_i$  — ранг наблюдения  $y_i$  в ряду  $Y$ . Коэффициент  $\rho$  принимает значения из отрезка  $[-1; 1]$ . Равенство  $\rho = 1$  указывает на строгую прямую линейную зависимость,  $\rho = -1$  — на обратную зависимость. Ранги для выборки расставляются в порядке возрастания  $x_i$ , при совпадении  $x_i$  и  $x_j$  рангу присваивается значение среднего арифметического между совпадающими значениями.

### 3. Практическая реализация

Все алгоритмы препроцессинга были реализованы с помощью библиотек PIL, OpenCV, SciPy и NumPy с использованием языка программирования и интерпретатора Python. OpenCV — библиотека алгоритмов компьютерного зрения.

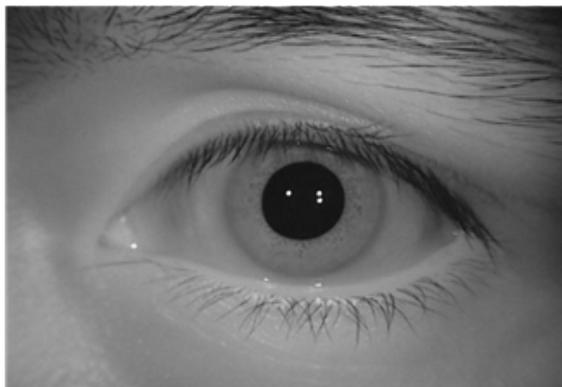


Рис. 2. Исходное изображение [8]



Рис. 3. Применение медианного фильтра к изображению



Рис. 4. Применение алгоритма CLAHE

**Реализация алгоритмов предобработки.** Для фильтрации изображения был использован медианный фильтр из библиотеки OpenCV. Медианный фильтр реализован с помощью функции `cv2.medianBlur(img,size)`, где `img` — исходное изображение, `size` — количество элементов для выборки.

После применения медианного фильтра изображение становится менее ярким, блики — более приглушёнными, удаляется часть шумов.

Следующая стадия — это повышение контрастности изображения, для этого были использованы две функции `clahe=cv2.createCLAHE(clipLimit, tileGridSize)`, где `clipLimit` — это специальный порог CLAHE, `tileGridSize` — размер области выравнивания, и `clahe.apply(img)`, где `img` — исходное изображение. Первая функция создаёт условия для работы алгоритма CLAHE, а вторая производит преобразования изображения.

После применения метода линии на изображениях становятся более чёткими, что в дальнейшем поможет выделить границы на изображении.

Для проведения процесса бинаризации переводим наши изображения в оттенки серого функцией `cv2.cvtColor(img,cv2.COLOR_GRAY2BGR)`. Далее

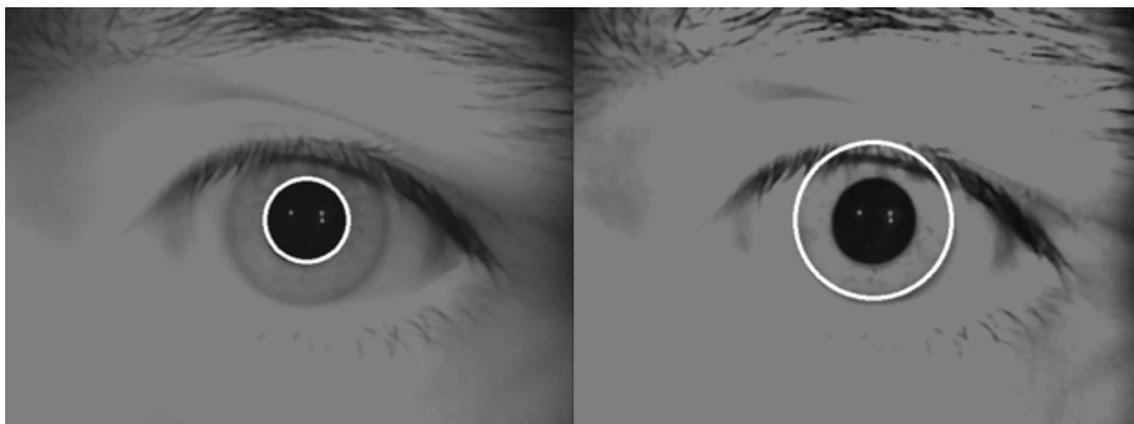


Рис. 5. Результат применения преобразования Хафа

проводим самую бинаризацию с помощью функции `cv2.threshold(img, thresh, maxValue, cv2.THRESH_TRUNC)`, где `img` — исходное изображение, `thresh` — порог бинаризации, `maxValue` — значение, которое получает пиксель выше порога, `cv2.THRESH_TRUNC` — тип бинаризации.

Для выполнения преобразования Хафа воспользуемся функцией `cv2.HoughCircles(img, cv2.CV_HOUGH_GRADIENT, dp, min_dist, param1, param2, minRadius, maxRadius)`, где `img` — это исходное изображение, `cv2.CV_HOUGH_GRADIENT` — используемый метод, `dp` — разрешение сумматора, используемое для детектирования центров кругов, `min_dist` — минимальная дистанция между центрами детектируемых кругов, `param1` — первый параметр (в зависимости от метода трансформации), `param2` — второй параметр (в зависимости от метода трансформации), `min_radius` минимальный радиус круга, `max_radius` — максимальный радиус круга.

Для нахождения зрачка глаза устанавливаем `minRadius=0` и `maxRadius=75`, для нахождения внешней границы радужной оболочки — `minRadius=85` и `maxRadius=130`.

После нахождения внешней границы радужной оболочки глаза обрезается изображение для удобства дальнейшей обработки. Необходимо найти посторонние объекты, которые могли попасть на радужку глаза (веки, ресницы). Для этого пользуемся детектором границ Canny. Для применения метода Canny берём функцию `cv2.Canny(img, threshold1, threshold2)`, где `img` — исходное изображение, `threshold1` и `threshold2` — левый и правый порог процедуры.

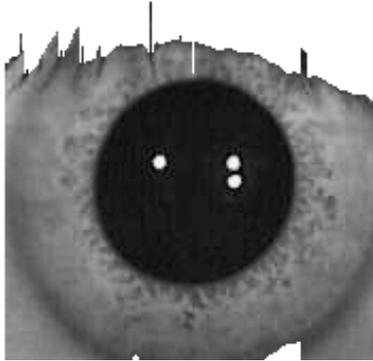


Рис. 6. Удаление лишних элементов на изображении



Рис. 7. Выделение общей границы

Для окончательного выделения границ накладываем полученные границы с помощью преобразования Хафа и детектора Саппу, и проводим общую границу.

Следующим шагом после нахождения границ является развёртка кольца оболочки. Для этой процедуры необходимо перевести координаты точек  $(x, y)$  из декартовой плоскости в координаты  $(\rho, \varphi)$  полярной по формулам

$$\begin{cases} \rho = \sqrt{x^2 + y^2}; \\ \varphi = \text{atan2}\left(\frac{y}{x}\right). \end{cases}$$

Расчёт значений начинается с пикселей, расположенных на внешней границе радужной оболочке глаза.

Вследствие перевода из декартовых координат в полярные теряются некоторые пиксели развёртки изображения. Для восстановления этих данных используется линейная интерполяция.



Рис. 8. Развёртка радужной оболочки

**Заполнение базы шаблонов.** Для сравнения образцов изображение необходимо привести к размеру 256x32 пикселя. `Image.resize(size, resample)`, где `size` — размер необходимого изображения, `resample` — тип преобразования (`Image.BICUBIC` — двойная сплайн-интерполяция).

**Идентификация заданного образца.** Для идентификации заданного образца реализуем несколько алгоритмов. Пусть на вход после проведения всей предварительной подготовки подан вектор  $x$  размером 8192 байта, каждая координата которого представлена значением от 0 до 255.

Алгоритм идентификации с помощью расстояния Хэмминга представлен в следующем виде.

Таблица 1. Результаты первого эксперимента (в процентах)

Метод	Процент правильных ответов
Расстояние Хэмминга	62
Скалярное произведение векторов	70
Коэффициент корреляции Спирмена	69

1. Посчитать расстояние Хэмминга для всех векторов, представленных в базе шаблонов.
2. Подать на выход номер объекта, расстояние которого было наименьшим. Этот объект и является результатом работы алгоритма.

Алгоритм идентификации с использованием скалярного произведения между векторами:

1. Посчитать косинус угла между заданным вектором и всеми векторами объектов в базе.
2. Подать на выход номер объекта, косинус которого был наибольшим. Этот объект и является результатом работы алгоритма.

Для улучшения алгоритма необходимо посчитать значения косинуса, сменяя в цикле значения векторов в базе на 32 бита. После всех подсчётов выбрать максимальное значение косинуса для классификации.

Алгоритм идентификации с использованием коэффициента корреляции Спирмена:

1. Вычислить ранги у заданного вектора и всех векторов объектов в базе.
2. Посчитать значение коэффициента корреляции Спирмена для заданного вектора и каждого вектора из базы.
3. Подать на выход номер объекта, коэффициент которого был наибольшим. Этот объект и является результатом работы алгоритма.

#### 4. Эксперимент

Для проверки работы метода использовалась база изображений глаз человека Casia-IrisV4 [8]. Были взяты изображения 10 человек, для каждого человека было дано 5 изображений глаза.

Было проведено два эксперимента для проверки алгоритма. В первом эксперименте алгоритму необходимо было решить проблему идентификации объекта, найти, с каким объектом у входного вектора больше всего совпадений.

Во втором эксперименте входной вектор сравнивался со всеми векторами базы, и в ответе показывался результат по каждому вектору базы (верификация объекта).

Для второго эксперимента необходимо вычислить значение порога  $p$ , при котором определяется принадлежность класса. Для этого воспользуемся следующими метриками:

1. TruePositive (TP) — количество образцов тестовой выборки, на которых

алгоритм выдаёт принадлежность к определённому объекту и достоверно этот образец принадлежит этому классу.

2. TrueNegative (TN) — количество образцов тестовой выборки, на которых алгоритм выдаёт непринадлежность к определённому объекту и достоверно этот образец не принадлежит этому классу.
3. FalsePositive (FP) — количество образцов тестовой выборки, на которых алгоритм выдаёт принадлежность к определённому объекту и достоверно этот образец не принадлежит этому объекту.
4. FalseNegative (FN) — количество образцов тестовой выборки, на которых алгоритм выдаёт непринадлежность к определённому объекту и достоверно этот образец принадлежит этому объекту.
5. Доля правильных ответов, выданных системой, на основе выше описанных параметров, вычисляется по формуле

$$accuracy = \frac{TP + TN}{TP + TN + FN + FP}.$$

6. Доля объектов, названных системой положительными и при этом действительно являющихся положительными:

$$precision = \frac{TP}{TP + FP}.$$

7. Доля объектов положительного класса, которые определила система, из всех объектов положительного класса:

$$recall = \frac{TP}{TP + FN}.$$

В таблицах 2, 3, 4 представлены результаты, полученные на основе тестов.

Выберем метод, который дал наилучший результат в первом эксперименте, и применим к нему сингулярное разложение, оставив только 16 первых компонент. Доля правильных ответов для первого эксперимента составляет 72 %.

Для второго эксперимента результат отражён в таблице 5.

На основе проведённых экспериментов можно сказать, что реализованные методы дают лучшие показатели при решении задачи идентификации.

## Заключение

В статье сделан обзор методов, применяемых для систем идентификации человека по радужной оболочке глаза: методы предварительной обработки изображения (фильтр Гаусса, медианный фильтр, бинаризация изображения, преобразование Хафа) и методы классификации объектов (метод опорных векторов, нейронные сети, метод Байеса, метод с использованием расстояния Хэмминга и т. д.). Реализованы методы предварительной обработки и методы идентификации объекта, основанные на расстоянии Хэмминга, скалярном произведении векторов и коэффициенте корреляции Спирмена, а также проведена их оценка.

В заключении можно сказать, что алгоритмы, реализованные в данной работе, можно усовершенствовать за счёт улучшения методов препроцессинга. Кроме того, для выделения признаков можно также использовать нейросети.

Таблица 2. Расстояние Хэмминга. Результаты эксперимента при  $p = 0,4$   
(в количестве выбранных объектов)

Объект	TP	TN	FP	FN	accuracy	precession	recall
1	0	38	187	20	0,15	0,07	0
2	3	48	177	17	0,2	0,01	0,145
3	16	3	222	4	0,07	0,06	0,8
4	2	69	156	18	0,28	0,01	0,1
5	11	50	175	9	0,24	0,05	0,55
6	14	15	210	6	0,11	0,06	0,7
7	8	47	178	12	0,22	0,04	0,4
8	18	13	212	2	0,12	0,07	0,9
9	7	69	156	13	0,31	0,04	0,35
10	14	9	216	6	0,09	0,06	0,7

Таблица 3. Скалярное произведение векторов. Результаты эксперимента при  $p = 0,55$   
(в количестве выбранных объектов)

Объект	TP	TN	FP	FN	accuracy	precession	recall
1	20	82	143	0	0,41	0,12	1
2	20	107	118	0	0,51	0,14	1
3	17	147	78	3	0,66	0,17	0,85
4	20	36	189	0	0,22	0,09	0,85
5	16	75	150	4	0,37	0,09	0,8
6	16	132	93	4	0,6	0,14	0,8
7	20	67	158	0	0,35	0,11	0,8
8	14	141	84	6	0,63	0,14	0,7
9	20	81	144	0	0,41	0,12	0,7
10	14	137	88	6	0,61	0,13	0,7

Таблица 4. Коэффициент корреляции Спирмена. Результаты при  $p = 0,5$   
(в количестве выбранных объектов)

Объект	TP	TN	FP	FN	accuracy	precession	recall
1	20	205	20	0	0,91	0,21	1
2	11	191	34	9	0,82	0,24	0,55
3	0	225	0	20	0,91	0,24	0,55
4	16	181	44	4	0,8	0,26	0,8
5	8	196	29	12	0,83	0,21	0,4
6	2	218	7	18	0,89	0,22	0,1
7	19	194	31	10	0,83	0,24	0,1
8	2	219	6	18	0,9	0,25	0,1
9	13	180	45	7	0,78	0,22	0,65
10	6	221	4	14	0,92	0,6	0,3

Таблица 5. Скалярное произведение векторов. Результат эксперимента при  $p = 0,55$   
(в количестве выбранных объектов)

Объект	TP	TN	FP	FN	accuracy	precession	recall
1	20	74	0	0	0,38	0,11	1
2	20	100	125	0	0,48	0,13	1
3	18	140	85	2	0,64	0,17	0,9
4	20	30	195	0	0,20	0,09	0,9
5	15	68	157	5	0,33	0,08	0,75
6	15	123	102	5	0,56	0,12	0,75
7	20	59	166	0	0,32	0,107	0,75
8	14	133	92	6	0,6	0,132	0,7
9	20	75	150	0	0,38	0,117	0,7
10	17	128	97	3	0,59	0,14	0,85

## Благодарности

Выражаем признательность Илье Дмитриевичу Сиганову за обсуждение результатов и ценные советы.

## ЛИТЕРАТУРА

1. Лавров Д.Н., Назарова Т.Ю. Алгоритмы предварительной обработки изображения для поиска границ радужной оболочки глаза // Молодёжь третьего тысячелетия: сборник научных статей. Омск : Изд-во Ом. гос. ун-та, 2017. С. 828–832.
2. Грищенко Н.П., Лавров Д.Н. Обзор методов идентификации человека по радужной оболочке глаза // Математические структуры и моделирование. 2014. №1(29). С. 43–64.
3. Бинаризация изображений. Распознавание образов для программистов. URL: <http://recog.ru/blog/applied/15.html> (дата обращения: 14.05.2016).
4. Дегтерева А., Вежнев В. Преобразование Хафа (Houghtransform) // Сетевой журнал «Компьютерная графика и Мультимедиа». URL: <http://sgm.computergraphics.ru/content/view/36> (дата обращения: 14.05.2018).
5. Матвеев И.А. Методы и алгоритмы автоматической обработки изображений радужной оболочки глаза: дис. ... канд. техн. наук. М. : ВЦ РАН, 2014.
6. Lopez P. SVM versus a monkey. Makeyourbets. URL: <https://quantdare.com/svm-versus-a-monkey> (дата обращения: 20.05.2017).
7. Zuiderveld K. Contrast limited adaptive histogram equalization. Graphics gems IV. Academic Press Professional, Inc., 1994. P. 474–485.
8. Biometrics Ideal Test. CASIA-IrisV4. URL: <http://biometrics.idealtest.org/dbDetailForUser.do?id=4> (дата обращения: 20.11.2015).

## COMPUTER MODELING OF IDENTIFICATION BY THE IRIS OF THE EYE BASED ON OPENCV

**T.Yu. Nazarova**

Student, e-mail: [tanyanazarova-94@yandex.ru](mailto:tanyanazarova-94@yandex.ru)

**D.N. Lavrov**

Ph.D. (Eng.), Associate Professor, e-mail: [lavrov@omsu.ru](mailto:lavrov@omsu.ru)

Dostoevsky Omsk State University, Omsk, Russia

**Abstract.** The results of recognition algorithm construction based on the use of the OpenCV library are given. The results of primary experiments on image recognition on a small base are presented.

**Keywords:** iris, identification, recognition.

*Дата поступления в редакцию: 09.09.2018*